IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION PAPERS

OF

DANIEL JOSEPH WOLFF

AND

GRAHAM ARTHUR MAKINSON

FOR

AUTHORISING AN ADDITIONAL COMPUTER PROGRAM MODULE FOR USE WITH A CORE COMPUTER PROGRAM

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to the field of data processing systems. More particularly, this invention relates to data processing systems in which it is desired to authorise an additional computer program module for use with an installed core computer program.

Description of the Prior Art

It is known to provide core computer programs that operate with one or more computer program modules associated therewith. One example of such an arrangement is an internet browser computer program that may operate with a number of plug-in modules that act to give additional functionality to the internet browser. Another example is an anti-virus scanning engine program that operates with a number of associated computer program modules that can provide virus definition data, latest virus definition data, mechanisms for decompressing or unpacking certain computer file types and functionality of a similar or associated nature.

A problem associated with such systems that allow additional computer program modules to be used in conjunction with a core computer program is that of ensuring that such additional computer program modules are properly secure, e.g. the additional computer program module is not a trojan that will damage the computer system if its use is authorised. One way of dealing with this is to provide a mechanism for additional computer program modules to identify their origin and then give the computer user the ability to either accept or decline that additional computer program module depending upon their perception of the risk. A more secure, but less flexible approach allows additional computer program modules from the provider of the core computer program to be used by checking a public key infrastructure signature associated with each additional computer program module to check that this matches with a predetermined checking algorithm within the core computer program that recognises genuine signatures from the core computer program provider.

While the approach of having the core computer program use a signature mechanism to check that additional computer program modules are secure, it does not permit the flexibility and extendibility of the plug-in approach used by internet browsers.

SUMMARY OF THE INVENTION

Viewed from one aspect the present invention provides a method of authorising an additional computer program module for use with a core computer program, said core computer program being installed, said method comprising the steps of:

reading module signature data associated with said additional computer program module;

reading core signature data and other signature data associated with said core computer program;

comparing said module signature data with said core signature data and said other signature data; and

refusing authorisation of said additional computer program module for use with said core computer program unless said module signature data matches at least one of said core signature data and said other signature data.

The invention recognises that the approach of coding into the core computer program itself the ability to check the signatures of additional computer program modules may be extended to accept additional computer program modules from parties other than the provider of the core computer program. In this way, a predetermined set of parties able to authorise additional computer program modules can be established that extends beyond the provider of the core computer program.

A strongly preferred feature of the invention is that the other signature data can be user signature data. In this way, a particular user may authorise an additional computer program module for use with their installed core computer program. Thus, a user may obtain an additional computer program module from a third party, or generate one themselves, and then apply their signature to it when they are happy that it is secure to use. This additional computer program module will then be recognised and authorised for use with the core computer program.

It will be appreciated that it would be possible for authorisation to be granted upon detection of either the core signature data or the other signature data. However, additional security may be provided when the system requires the presence of both signatures before authorising use of an additional computer program module.

It will be appreciated that the signature data could take many different secure data forms. However, the use of public key infrastructures signatures is particularly well suited to the present invention.

The signature data may be associated with the additional computer program modules or core computer program in various ways, such as stored in an accompanying table, but is preferably embedded within the item it seeks to verify as this lends itself to greater tamper resistance.

Whilst the core computer program could take many forms, particularly preferred embodiments of the invention are ones in which the core computer program is an anti-virus computer program. Anti-virus computer programs are widely used and because of their critical nature in computer systems are carefully managed and controlled. A large organisation will often have a defined anti-virus policy requiring the installation and running of an anti-virus computer program on every computer system within that organisation. Furthermore, it is common for mechanisms and procedures to be in place to permit the regular and rapid updating of these anti-virus computer programs with the very latest virus definition data and scanning engines. This infrastructure for the distribution of virus definition data and scanning engines can be utilised for the distribution of additional computer program modules that have a role outside of the anti-virus role without requiring significant extra effort in establishing a distribution mechanism for those new additional computer program modules.

As preferred examples of the functionality that may be provided by the additional computer program modules, modules could be provided that act to install a computer program patch, install a new piece of software or install system monitoring software for reporting upon the hardware and software configuration of a computer system to a central source. These modules may be distributed using the mechanisms normally employed for anti-virus system updates and yet achieve functionality outside of that normally associated with anti-virus systems. In this context, it is important that the additional computer program modules should

be properly authorised for use as otherwise the anti-virus system could itself be utilised to spread damaging additional computer program modules.

The generation of properly authorised additional computer program modules could take place in various different ways. A tool for permitting user signature data to be added to computer program modules could be licensed to software developers by the producer of the core computer program together with appropriate tools for compiling computer program modules to interact with that core computer program. The software writer would then distribute computer program modules with appropriate signatures authorised for respective users upon payment from that user for use of the new computer program module.

An alternative way in which a new computer program module may be authorised would be that it would be signed by the user and then sent together with a fee to the core computer program provider for signing by them. The core computer program provider would then return the doubly signed computer program module to the user so that it could then be used and would forward on a proportion of the fee received from the user to the writer of the computer program module.

A further possibility would be that a writer of computer program modules would be sold a tool for signing their own signature on those modules with the user then having the ability to effectively also add their signature to those modules, either individually as they are created, or on mass for any modules from that provider, such that they may then be authorised for use upon the particular core computer program.

Viewed from another aspect the invention also provides apparatus for authorising an additional computer program module for use with a core computer program, said core computer program being installed, said apparatus comprising:

first reading logic operable to read module signature data associated with said additional computer program module;

second reading logic operable to read core signature data and other signature data associated with said core computer program;

a comparator operable to compare said module signature data with said core signature data and said module signature data; and

authorisation refusal logic operable to refuse authorisation said additional computer program module for use with said core computer program unless said module signature data matches at least one of said core signature data and said other signature data.

Viewed from a further aspect the invention provides a computer program product carrying a computer program for controlling a computer to authorise an additional computer program module for use with a core computer program, said core computer program being installed, said computer program comprising:

first reading code operable to read module signature data associated with said additional computer program module;

second reading code operable to read core signature data and other signature data associated with said core computer program;

comparison code operable to compare said module signature data with said core signature data and said module signature data; and

authorisation refusal code operable to refuse authorisation said additional computer program module for use with said core computer program unless said module signature data matches at least one of said core signature data and said other signature data.

The above, and other objects, features and advantages of this invention will be apparent from the following detailed description of illustrative embodiments which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 schematically illustrates a core computer program operating in conjunction with a plurality of additional computer program modules;

Figure 2 is a flow diagram illustrating how the core computer program may validate the signatures of the computer program modules attached to it upon start-up;

Figures 3, 4 and 5 give examples of functionality that may be provided by additional computer program modules associated with a core computer program;

Figure 6 illustrates how the core computer program may be customised for a particular user to enable that user to authorise computer program modules;

Figure 7 gives more detail of one example of how the signatures associated with a computer program module may be checked;

Figures 8 and 9 illustrate the stages by which a computer program module may make its way from being written to being deployed by an end user;

Figure 10 illustrates a further way in which signature data may be checked by the core computer program.;

Figure 11 illustrates a further way in which a computer program module may progress from the writer to the end user; and

Figure 12 schematically illustrates a general purpose computer that may be used to implement the techniques described above.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 illustrates a core computer program 2 that in this example is an anti-virus computer program. Virus definition data 4, 6 is associated with the core computer program 2. In addition to the computer program modules 4, 6 that are the virus definition data, additional computer program modules #0, #1 and #2 are provided to allow the core computer program 2 to deal with different sorts of compressed computer files or other new processing requirements that may be encountered or developed. As an example, if it is decided to support scanning of computer files compressed with a new type of compression algorithm , then an additional computer program module may be produced and distributed so as to couple to the core computer program 2 to provide this new functionality.

In order to improve the security of the system, each of the computer program modules includes one or more signatures in the form of public key infrastructure signatures. The core computer program 2 looks for and checks the correctness of these signatures before authorising the associated computer module to interact with the core computer program 2.

Figure 2 illustrates the behaviour of the core computer program 2 upon start-up. At step 8, the core computer program is started, e.g. at boot of the computer system or when the anti-virus system is initiated for an on-demand scan. At step 10, the first computer program module associated with the core computer program 2 is selected. At step 12, the PKI signature within the selected computer program module is identified and checked for authenticity. If this check is successful, then step 14 registers that module for authorised use by the core computer program 2, and if appropriate runs the computer program module. If the check of the PKI signature as step 12 failed, then step 16 generates a report to the user of this failure and potential security problem as well as bypassing step 14. At step 18, a check is made as to whether or not there are any more computer program modules to potentially register. If there are more modules to register, then the next of these is selected at step 20 and processing is returned to step 12, otherwise the registration process terminates.

Figure 3 illustrates an example of a computer program module that serves to act as a system monitor. This computer program module is automatically run when the core computer program 2 registers it and serves to report the hardware and software configuration of the computer system to a central source. At step 22, the computer program module collects information concerning the hardware configuration of the system. As the anti-virus core computer program 2 has low level access to the computer system upon which it operates, it is typically able to generate full and accurate information regarding the hardware configuration of the system, e.g. it can give a true indication of the system memory rather than potentially including virtual memory as may be reported by higher level systems. At step 24, information concerning the installed software on the computer system may be collected, e.g. from an examination of the program registry. At step 26, a report of the collected information is sent to a central source, such as a network management computer.

Figure 4 illustrates a further example of a computer program module. This computer program module is used for the distribution of a patch to an already installed computer program. The anti-virus core computer program 2 is particularly well suited to this use as it already has highly efficient mechanisms for searching and identifying particular computer files upon a computer system. If one considers the computer file(s) that require to be patched as equivalent to virus damaged file(s), then these may be identified and effectively repaired by being replaced by

the updated patched versions of those computer files. The patching computer program module may also add any new files required and delete redundant files.

At step 28, a scan is made for the computer files that it is required to patch. These computer files are known and accordingly a simple search can be made for the file name or a small portion of the content of these files. When the files have been found, they are replaced at their existing location at step 30 with the new versions of those files in a manner equivalent to the repairing of a computer virus damaged file. At step 32, any new additional files required by the patch are added and at step 34 any old files now known to be redundant are moved.

Figure 5 illustrates a computer program module that may be used to install new software on a computer system. At step 36, a scan is made of the computer system to see if the new application software is already installed upon the system. If the new software is already installed, then processing terminates. If the new software is not present, then step 38 serves to add the computer files required for the new software in the appropriate location(s) and make any other changes to the system, such as registry entries that may be required. At step 40, a check is made to see if any old versions of the software are present and, if these are found, then they are deleted at step 44 (optionally following a confirmation input from a user) before processing terminates.

Figure 6 illustrates how the core computer program 2 may be modified upon its installation to enable it to check for a user signature specific to that installation. At step 44 an installation wizard is started upon the system on to which it is desired to install the core computer program. At step 46, the user is prompted to enter licence and user information. At step 48, this information is sent, e.g. via an internet link, to the core provider. At step 50, the core provider uses the received information to generate a user specific patch that will allow the core computer to recognise a user specific PKI signature applied to any additional computer program modules. This user specific patch is then returned to the user. At step 52, the user applies the received patch together with the other parts of the core computer program to their computer system.

Figure 7 illustrates one example of how the signature associated with a computer program module may be checked. At step 54, the core computer program checks for the presence of the core provider's signature. If this is not present, then that computer program module is marked as a fail at step 56 and processing terminated. If the core provider signature is

identified, then step 58 searches for any additional signatures. If additional signatures are found, then processing proceeds to step 60 to check that the additional signature matches the user signature and accordingly that the additional module has been authorised by that user to be installed upon the system concerned. If the user signature is not detected at step 58, then step 60 is bypassed. If the check at step 60 is passed, then the computer program module is marked as passing the authorisation test at step 62, otherwise processing proceeds to step 56.

Figure 8 schematically illustrates the interactions that occur between a user, a provider of a core computer program and a writer of an additional computer program module. The module provider can create a new computer program module that they may advertise the new module for sale in the normal way to attract users.

When a user has decided to purchase the new module, they may obtain a copy of the completely unsigned module directly from the module provider and then apply their own signature to it. The user then passes this module bearing the user's signature together with a payment to the core computer program provider. The core computer program provider checks that the user is properly registers as a user of their core computer program and that the payment is valid. If both of these conditions are met, then the core computer program provider can then apply their own signature to the module such that the module now bears both the user's signature and the core provider's signature. This doubly signed module is thereby activated in a system that requires both the user signature and the core provider signature to be present. The activated module is returned to the user by the core provider.

The core provider may retain a portion of the payment received from the user and pass on the remainder of the payment to the module provider. In this way, the module provider and the core provider both receive a per registration payment for use of the module and use of the core provider's mechanism for authorising that module.

Figure 9 illustrates an alternative way by which a module may make its way from the module provider to the user. The module provider can create a new module and then submit it together with an appropriate one time payment to the core provider for the core provider to mark it with an appropriate signature. This could be the core provider's own signature, but is preferably a signature specific to the module provider concerned. The module provider then has a signed module which they may advertise to attract users.

When a user decides to buy the module, they send a payment to the module provider who when the payment has been validated returns the signed module to the user. In this case, the user's core computer program needs to check the signature on the module to check it is an authorised signature. If the signature is one that is specific to the module provider, then the core computer program can effectively require the user to countersign the module thereby indicating that they are happy to accept the module from the module provider identified by the signature on the module. The core computer program may keep a list of module providers who have been previously authorised by the user such that the user's signature effectively also becomes associated with any modules provided by module providers indicated within the list as authorised by the user. As illustrated in Figure 9, the module provider may also return a royalty payment on a per use basis by a user to the core provider.

Figure 10 illustrates an alternative way in which the core computer program may check the signatures. In this example, step 64 checks for the presence of either the core provider signature or the user's signature and authorises the module if at least one of them is found. Thus, it is sufficient for an individual one of the core provider's or user's signatures to be found. Step 66 marks the module as failing authorisation and step 68 marks the module as passing authorisation.

Figure 11 illustrates a further way in which a module may make its way from the module provider to a user. A module provider writes a new module and then obtains an appropriate signature tool from the core provider upon payment to the core provider of a fee. This signature tool allows the module provider (possibly for a limited time) to apply signatures to their module authorising specific users.

When a user decides to buy the new module they send their user specific information together with payment to the module provider. The module provider uses the user information to generate a user signature using the tool they have purchased once the user's payment for the module has been verified. The module thus signed with the user's signature is then returned to the user where it will be recognised as authorised by the core computer program of that user and accordingly deployed. As previously described, a royalty payment may also be made on a per user basis to the core provider when the tool is used to apply a user signature.

Figure 12 illustrates a general purpose computer 70 of the type which may be used to implement the above described techniques. The general purpose computer 70 comprises a central processing unit 72, a read only memory 74, a random access memory 76, a hard disk drive 78, a display driver unit 80 and display 82, a user input/output unit 84 and a keyboard 86 and mouse 88 and a network link unit 90 which are all linked together via a common bus 92. In operation the central processing unit 72 executes computer program instructions that may be stored within one or more of the read only memory 74, the random access memory 76 or the hard disk drive 78. Computer program instructions may also be downloaded via a computer network using the network link unit 90. The computer hardware may also be considered when programmed by the computer program instructions to provide physical logic for yielding the desired functions. The computer program instructions may be loaded into the general purpose computer 70 using a distribution medium such as a compact disk or floppy disk. Alternatively, the computer program instructions could be carried by the network to which the general purpose computer 70 is connected and downloaded into the general purpose computer 70.

Although illustrative embodiments of the invention have been described in detail herein with reference to the accompanying drawings, it is to be understood that the invention is not limited to those precise embodiments, and that various changes and modifications can be effected therein by one skilled in the art without departing from the scope and spirit of the invention as defined by the appended claims.